



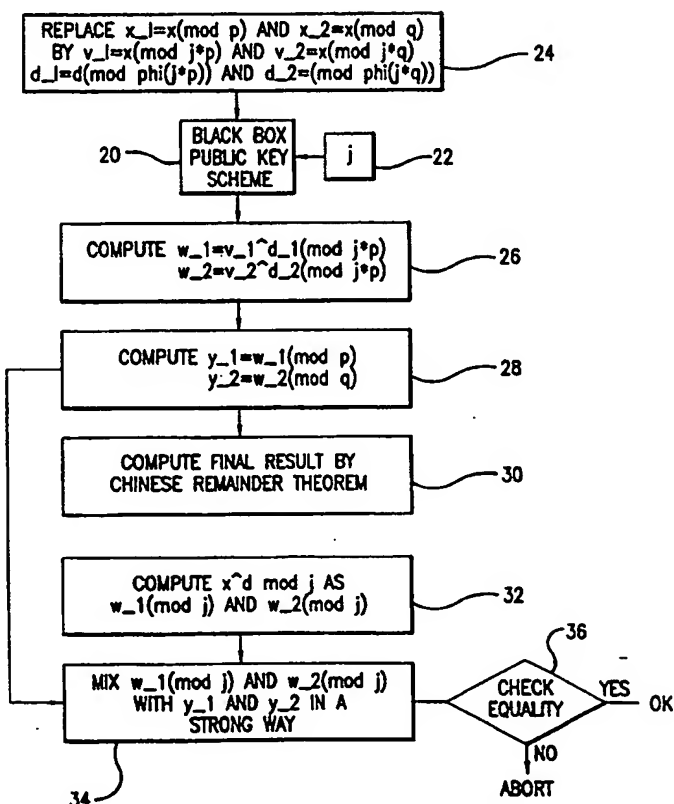
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT) -

<b>(51) International Patent Classification <sup>6</sup>:</b> <b>H04L 9/30</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 98/52319</b> <b>(43) International Publication Date:</b> 19 November 1998 (19.11.98)
<b>(21) International Application Number:</b> PCT/US98/09593 <b>(22) International Filing Date:</b> 12 May 1998 (12.05.98) <b>(30) Priority Data:</b> 08/854,464 12 May 1997 (12.05.97) US <b>(71) Applicant:</b> YEDA RESEARCH AND DEVELOPMENT CO. LTD. [-/IL]; Weizmann Institute of Science, P.O. Box 95, 76100 Rehovot (IL). <b>(71) Applicant (for UG only):</b> FLEIT, Lois [US/US]; 520 Brickell Key Drive #A201, Miami, FL 33131 (US). <b>(72) Inventor:</b> SHAMIR, Adi; 43C Gordon Street, Rehovot (IL). <b>(74) Agent:</b> FLEIT, Martin; Lott & Friedland, Suite 555, 255 Alhambra Circle, Coral Gables, FL 33134 (US).		<b>(81) Designated States:</b> AL, AM, AU, AZ, BA, BB, BG, BR, BY, CA, CN, CU, CZ, EE, FI, GE, GH, GM, GW, HU, ID, IL, IS, JP, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LV, MD, MG, MK, MN, MX, NO, NZ, PL, RO, RU, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

**(54) Title:** IMPROVED METHOD AND APPARATUS FOR PROTECTING PUBLIC KEY SCHEMES FROM TIMING AND FAULT ATTACKS

**(57) Abstract**

Improved methods and apparatus for protecting public key schemes based on modular exponentiation (including RSA and Diffie-Hellman) from indirect cryptanalytic techniques such as timing and fault attacks. Known methods for making the implementation of number-theoretic schemes resistant to such attacks typically double their running time, whereas the novel methods and apparatus described in this patent add only negligible overhead. This improvement is particularly significant in smart card and software-based implementations, in which the modular exponentiation operation is quite slow, and doubling its time may be an unacceptable solution.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

# IMPROVED METHOD AND APPARATUS FOR PROTECTING PUBLIC KEY SCHEMES FROM TIMING AND FAULT ATTACKS

## FIELD OF INVENTION

The present invention relates to novel techniques, methods and apparatus, for making number-theoretic public key schemes (including encryption schemes, signature schemes, identification schemes, key management schemes, etc.) resistant to timing and fault attacks.

## BACKGROUND OF INVENTION

### 1. Introduction.

The simplest attack on a given cryptosystem is to exhaustively search for the key. There are many variants of this attack (known ciphertext, known cleartext, chosen cleartext, etc.), but they are all based on a procedure which tries the keys one by one until the correct key is encountered. If the key consists of  $n$  random bits, the expected running time of this procedure is  $2^{(n-1)}$ . This attack can be easily foiled by using a large enough  $n$  (e.g.,  $n > 100$ ).

To attack cryptosystems with large keys, cryptanalysts try to find mathematical or statistical weaknesses which reduce the size of the search space (preferably to 1). Although many techniques and results are classified for national security reasons, it is safe to assume that it is increasingly difficult to find such weaknesses in modern schemes designed by experienced cryptographers and implemented on high speed microprocessors.

To successfully attack strong cryptosystems, the cryptanalyst must use indirect techniques. This is best done when the cryptanalyst is either in close physical

proximity to the cryptographic device, or has it under his complete control. The cryptographic device is assumed to be a black box which contains a known algorithm and an unknown key. The cryptanalyst cannot open this box and read its key, but he can observe its behavior under various circumstances.

One of the best known examples of such an indirect attack is TEMPEST, which tries to deduce the key by analyzing electromagnetic radiation emanating from the black box during the computation of the ciphertext. Techniques for applying and preventing such attacks have been extensively studied for more than 50 years, and by now this is a well understood problem.

Two powerful indirect attacks were discovered and published recently: In December 1995, P. Kocher, "Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks," technical report, 12/7/95, described a timing attack, and in September 1996, D. Boneh, R.A. Demillo and R.J. Lipton, "On the Importance of Checking Computations," technical report, 9/25/96 (an extended version appears in the Proceedings of Eurocrypt 97, May 1997) described a fault attack. Both attacks were originally designed for and are most successful against public key schemes based on number theoretic principles, such as RSA, but they were later extended to classical cryptosystems as well (e.g., by E. Biham and A. Shamir, "A New Cryptanalytic Attack on DES," technical report, 10/18/96. An extended version appears in the Proceedings of Crypto 97, August 1997).

Such attacks are particularly useful when the scheme is implemented on a smart card, which is distributed by a bank, computer network, cellular phone operator, or

pay-TV broadcaster to its customers. Hackers do not usually have the financial and technical resources required to read the contents of the key registers inside the smart card, but they have complete control on the input/output, clock, reset, and power connections of the smart card. They can carefully measure the duration of the various operations, how much power they consume, what happens when the computation is interrupted or carried out under abnormal operating conditions, etc. Since the tests are carried out in the privacy of the customer's home, the card manufacturer cannot prevent them or even learn about their existence.

## 2. Timing Attacks.

Timing attacks are based on the assumption that some of the basic operations carried out during the cryptographic calculation require a non-constant amount of time which depends on the actual values being operated upon. This implies that some information about these unknown intermediate values leaks out by measuring the length of the cryptographic computation. If these intermediate values are computed from known cleartext bits and unknown key bits by a known cryptographic algorithm, the attacker can try to use the leaked intermediate values to deduce the key.

The main difficulty in carrying out this attack is that the attacker knows only the total amount of time required to carry out the cryptographic computation, but not the timing of individual computational steps. Kocher's main contribution is in developing an efficient technique for handling this difficulty in many cases of practical interest.

For the sake of concreteness, we describe Kocher's attack on the RSA

cryptosystem. The black box is assumed to contain a publicly known modulus  $n$  and a secret exponent  $d$ . Given an input number  $x$ , the box performs the modular exponentiation  $x^d \pmod{n}$  by using the standard square-and-multiply technique. In this description, the symbol " $^$ " is exponentiation and the symbol " $_$ " is a subscript. The result (which can be the decryption of the ciphertext  $x$ , the signature of the message  $x$ , or the response to a random identification challenge  $x$ , depending on the application) is sent out as soon as it is produced, and thus the attacker can measure the total number of clock cycles taken by all the modular multiplications.

Standard implementations of modular multiplication require a non-constant amount of time, since they skip multiplication steps involving leading zeroes, and reduction steps when the result is smaller than the modulus. The attacker chooses a large number of random inputs  $x$ , and measures the actual timing distribution  $T_0$  of the modular exponentiation operation carried out by the black box. He then measures for each  $x$  (by computer simulation, using his knowledge of how the scheme is implemented) the precise timing of an initial square-only operation, and separately, the precise timing of an initial square-and-multiply operation. The result is a pair of timing distributions  $T_1$  and  $T_2$ , which are not identical. All the cryptographic computations carried out in the black box use the same exponent  $d$ , and its first bit determines which one of the two computed distributions  $T_1$  and  $T_2$  is the initial part of the experimentally computed  $T_0$ . The timing of the remaining steps of the computations can be assumed to be a random variable  $R$ , which is normally distributed and uncorrelated with either  $T_1$  or  $T_2$ . Since  $T_0$  is either  $T_1+R$  or  $T_2+R$ , the attacker can decide which case is more

likely by finding which one of the two distributions  $T_0-T_1$  and  $T_0-T_2$  has a lower variance.

After finding the first bit of the secret exponent  $d$ , the attacker knows the actual inputs to the second computational step, and thus he can apply the same technique (with properly modified experimental and simulated timing distributions  $T'_0$ ,  $T'_1$ , and  $T'_2$ ) to find the second bit of  $d$ . By repeating this procedure about 1000 times, he can compute all the bits of  $d$ , and thus break the RSA scheme.

A similar timing attack can be applied to any cryptographic scheme in which the black box raises all its inputs  $x_1, x_2, \dots$  to the same secret power  $d$  modulo the same known  $n$  (which can be either a prime or a composite number). For example, in one of the variants of the Diffie-Hellman key distribution scheme, all the users agree on a prime modulus  $n$  and on a generator  $g$  of the multiplicative group  $Z^{*}_n$ .

Each user chooses a random secret exponent  $d$ , and computes  $y = g^d \pmod{n}$  as his public key. To establish a common secret key with another user, the first user sends out his public key  $y = g^d \pmod{n}$ , and receives a similarly computed public key  $x = g^e \pmod{n}$  from the other user. Their common cryptographic key is  $z = g^{(d \cdot e)} \pmod{n}$  which the first user computes by evaluating  $x^d \pmod{n}$ . When the first user communicates with several parties, he raises several known values  $x_1, x_2, \dots$  to the same secret power  $d$  modulo the same known modulus  $n$ . By measuring the timing of sufficiently many such computations, the attacker can determine  $d$  and thus find all the cryptographic keys  $z_1, z_2, \dots$  employed by that user.

The timing attack has to be modified if the computation of  $x^d \pmod{n}$

for a composite modulus  $n=p*q$  is carried out by computing  $x^d \pmod p$ ,  $x^d \pmod q$ , and combining the results by the Chinese Remainder Theorem (CRT). This is a common way of making the computation about 4 times faster when the factorization of  $n$  is known. The problem for the attacker is that he does not know the secret factors  $p$  and  $q$  of the public modulus  $n$ , and thus cannot simulate the timing distributions  $T_1$  and  $T_2$ . Kocher's solution is to concentrate on the first step of the CRT computation, in which the input  $x$  is reduced modulo  $p$ . If  $x$  is smaller than  $p$ , no modular reduction is required, and thus the computation is considerably faster than when  $x$  is larger than or equal to  $p$ . The attacker thus presents to the black box a large number of inputs  $x$  which are very close to each other, and uses the average time of such computations to decide whether these  $x$ 's are above or below  $p$ . A decision procedure for this question can be repeatedly used to find the precise value of  $p$  by binary search.

Shortly after the discovery of this attack, researchers tried to develop implementations which are immune to it. The simplest idea is to make sure that all the cryptographic operations take exactly the same amount of time, regardless of the values of the cleartexts and keys. However, achieving this is surprisingly difficult for the following reasons:

- (a) In many cases, the implementor wants to run the same algorithm in software on different (and perhaps unknown) machines. An implementation which is constant time on one microprocessor may be variable time on another microprocessor or even on an enhanced version of the same microprocessor.



- (b) On a multitasking machine the running time may depend on the amount of free memory, the cache hit rate, the number of external interrupts, etc. This can change a constant time implementation under one set of circumstances into a variable time implementation under another set of circumstances.
- (c) If the implementor tries to use a real time clock to force all the computations to take the same amount of time, he must slow all of them down to their worst cases. Since he cannot use any input-dependent optimization technique, the implementation is likely to be unacceptably slow.

The best protective technique proposed so far against Kocher's timing attack on modular exponentiation is to replace each input  $x$  by a modified version  $y = x \cdot r \pmod{n}$  where  $r$  is a secret random number between 1 and  $n-1$ . To compute  $x^d \pmod{n}$ , the program computes  $y^d \pmod{n}$  and  $r^d \pmod{n}$ , and then uses the multiplicative property of modular exponentiation to compute  $x^d \pmod{n}$  as  $y^d / r^d \pmod{n}$ . Since both  $y$  and  $r$  are unknown, the attacker cannot simulate these computations in order to find the successive bits of  $d$  in the non-CRT computation, and cannot perform binary search in the CRT version of the computation. Unfortunately, this randomization technique doubles the expected running time of the computation.

### 3. Fault Attacks.

Fault attacks try to introduce errors into the cryptographic computation, and to identify the key by analyzing the mathematical and statistical properties of the

erroneously computed results. Among the many techniques suggested so far for introducing such errors are the use of ionizing radiation, unusual operating temperatures, power and clock glitches, and laser-based chip microsurgery. Some of the attacks are differential (i.e., they carry out both a correct and an erroneous computation with the same input and analyze their differences), while other attacks just use the erroneous results.

The original fault attack on public key cryptosystems was described in Boneh, Demillo and Lipton, and required several cryptographic computations. We now describe an improved version of this attack, due to Arjen Lenstra, which requires a single faulty computation. We assume that the black box uses the RSA scheme to sign a given message  $x$ . The computation of  $x^d \pmod{n}$  is carried out with the CRT method by first reducing  $x$  modulo  $p$  and  $q$  to get  $x_1$  and  $x_2$ , then computing  $y_1 = x_1^d \pmod{p}$  and  $y_2 = x_2^d \pmod{q}$ , and finally combining  $y_1$  and  $y_2$  to get the signature  $y \pmod{n}$  with the CRT method. We assume that a single error is introduced at a random time during this computation by applying mild physical stress to the black box. Without loss of generality, we can assume that the error was introduced during the computation of  $x_1^d \pmod{p}$ , and thus instead of getting the correct  $y_1$ , the box computed an erroneous  $y'_1$ . When  $y'_1$  and  $y_2$  are combined by the CRT method, the box computes an incorrect signature  $y'$  which is provided to the attacker.

The main observation is that the attacker knows the signature verification exponent  $e$ , for which  $y^e = x \pmod{n}$ . Due to the error,  $y'^e - x$  is non-zero mod  $p$ , but zero mod  $q$ , and thus it is a multiple of  $q$  which is not a multiple of  $n$ . The attacker can

thus factor  $n$  by computing the greatest common divisor of  $y'^e \pmod n$  and  $n$ , which is an easy computation.

To protect cryptographic schemes against fault attacks, Boneh, Demillo and Lipton recommend that each computation should be carried out twice (preferably by different algorithms). If any discrepancy is found between the two results, the box should not output anything. This provides strong protection from random faults (which are unlikely to affect the two computations in an identical way), but it slows down the computation by a factor of 2. Such a slowdown is particularly noticeable in smart card implementations of public key schemes, which are quite slow to begin with.

### SUMMARY OF THE INVENTION

The present invention relates to method and apparatus for providing protective techniques for public key schemes, which provide strong protection against the described timing and fault attacks without incurring the twofold slowdown made necessary by the previously known protective techniques.

The first technique is designed to protect non-CRT implementations of public key schemes against timing attacks. It is applicable to the RSA cryptosystem, RSA digital signature scheme, Diffie-Hellman key distribution scheme, and any other number theoretic scheme in which the black box raises a known input  $x$  to a fixed secret exponent  $d$  modulo a public  $n$  whose factorization is known to the black box.

The second technique is designed to protect CRT-based implementations of public key schemes from both timing and fault attacks. The main problem is how to verify the correctness of the computations of  $x_1^d \pmod p$  and  $x_2^d \pmod q$

without repeating them a second time (or verifying each step separately, which again doubles the running time). The invention provides a novel error-detection technique for such number theoretic computations which is much more efficient.

### BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 shows schematically the method and apparatus of the invention as it relates to a first technique designed to protect non-CRT implementations of a public key scheme against timing attacks.

Fig. 2 shows schematically the method and apparatus of the invention as it relates to a second technique designed to protect CRT-based implementations of a public key scheme against both timing and fault attacks.

### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to the drawings, preferred embodiments will now be described in detail. Since public key schemes and computer hardware and software implementation are well known to those of skill in the art, no description of same is deemed necessary to a full, concise and exact understanding of the present invention.

We now describe two novel protective techniques for public key schemes, which provide strong protection against the described timing and fault attacks without incurring the twofold slowdown made necessary by the previously known protective techniques.

The first technique, shown in Fig. 1, is designed to protect non-CRT implementations of public key schemes against timing attacks. It is applicable to the RSA cryptosystem, RSA digital signature scheme, Diffie-Hellman key distribution scheme, and any other number theoretic scheme in which the black box 10 raises a

known input  $x$  to a fixed secret exponent  $d$  modulo a public  $n$  whose factorization is known to the black box.

The main observation is that for each  $n$  there exists a number  $t = \phi(n)$ , box 12, such that for any  $x$  between 1 and  $n$  which is relatively prime to  $n$ ,  $x^t = 1 \pmod{n}$ . This  $\phi$  is called Euler's totient function: when  $n$  is prime,  $\phi(n) = n-1$ , and when  $n = p \cdot q$ ,  $\phi(n) = (p-1) \cdot (q-1)$ . The implementation can thus replace the computation of  $x^d \pmod{n}$  by the computation of  $x^{(d+i \cdot t)} \pmod{n}$  for any integer  $i$  without changing the computed result, since  $x^{(d+it)} = (x^d) \cdot (x^t)^i = (x^d) \cdot 1^i = x^d \pmod{n}$ , box 14.

This equality can be shown to be true even when  $x$  is not relatively prime to  $n$ , but this case is unlikely to arise in practice. Note also that any integral multiple of  $\text{GCD}((p-1), (q-1))$  can replace  $\phi(n)$  in our choice of  $t$ .

The proposed protection technique for public key schemes based on the modular exponentiation operation is to choose a new random secret  $i$  in each computation, and to replace the computation of  $x^d \pmod{n}$  by the computation of  $x^{(d+i \cdot t)} \pmod{n}$  where  $t$  is the precomputed value of  $\phi(n)$ . Since the bits of these  $(d+i \cdot t)$  for the various  $i$  are different, each exponentiation uses a different sequence of square-and-multiply steps, and thus the attacker cannot use Kocher's timing attack to analyze the timing distribution of several exponentiations, even though all of them compute the same  $d$ -th power of their inputs.

The efficiency of this technique is based on the fact that  $t$  is always smaller than  $n$ , and the randomizing element  $i$  can be chosen as a relatively small number. If  $n$

and  $d$  are 1024 bit numbers, and  $i$  is a random 32 bit number (which are the currently recommended sizes),  $d+i*t$  is a 1056 bit number. The process of raising the input  $x$  to the 1056 bit power  $d+i*t$  requires only 3% more square-and-multiply operations than the process of raising  $x$  to the original 1024 bit power  $d$ . This is much better than the alternative randomization technique described earlier, which doubles the running time.

The second technique, shown in Fig. 2, is designed to protect CRT-based implementations of public key schemes from both timing and fault attacks. The main problem is how to verify the correctness of the computations of  $x_1^d \pmod{p}$  and  $x_2^d \pmod{q}$  without repeating them a second time (or verifying each step separately, which again doubles the running time). We now describe a novel error-detection technique for such number theoretic computations which is much more efficient.

In each computation, the black box 20 chooses a new random integer  $j$  (the recommended size of  $j$  is also 32 bit), box 22. Instead of computing  $x_1 = x \pmod{p}$  and  $x_2 = x \pmod{q}$  followed by  $y_1 = x_1^d \pmod{p}$  and  $y_2 = x_2^d \pmod{q}$ , the box computes  $v_1 = x \pmod{j*p}$ ,  $v_2 = x \pmod{j*q}$ ,  $d_1 = d \pmod{\phi(j*p)}$ , and  $d_2 = d \pmod{\phi(j*q)}$ , box 24, followed by  $w_1 = v_1^{d_1} \pmod{j*p}$  and  $w_2 = v_2^{d_2} \pmod{j*q}$ , box 26.

The main observation is that from  $w_1$  and  $w_2$  it is easy to derive  $y_1$  and  $y_2$  by further reductions (namely,  $y_1 = w_1 \pmod{p}$  and  $y_2 = w_2 \pmod{q}$ ), box 28, and thus it is easy to compute the final result  $y$  by the Chinese remainder Theorem, box 30. However, we can also obtain the value of  $x^d \pmod{j}$  in two different ways: as  $w_1 \pmod{j}$  and as  $w_2 \pmod{j}$ , box 32. We can now use the equality of these two

values (which were obtained from the two halves of the computation, and mixed with the derivation of  $y_1$ ,  $y_2$  in a very strong way), box 34, as a test of correctness: In a faultless computation the two values will always be the same, whereas in a faulty computation (with random faults) the probability that the two values will be the same is about  $1/(2^{32})$ , see decision box 36 in which an abort is commanded for a faulty computation. This error detection technique is thus sufficient for any application in which the total number of modular exponentiations is significantly smaller than  $2^{32}$  (about 4 billion).

The overall time complexity of this implementation is higher by a few percent than the time complexity of standard implementations, since the exponentiations are carried out modulo  $512+32=544$  bit moduli  $j \cdot p$  and  $j \cdot q$  instead of 512 bit moduli  $p$  and  $q$ . However, this is much faster than repeating each exponentiation a second time to verify its correctness.

An additional benefit of this randomization technique is that it also provides protection from timing attacks at no extra cost. Kocher's original attack on CRT-based implementations concentrates on the initial modular reduction (mod  $p$ ), and uses binary search to find increasingly accurate approximations of  $p$  from multiple computations.

By using our proposed technique, each computation uses a different modulus  $j \cdot p$  in its initial reduction step, and thus the attacker cannot refine his knowledge of the modulus by analyzing a large number of exponentiations.

There are many optimizations and variations of this technique, which

should be obvious to anyone skilled in the art. For example, it is possible to impose additional restrictions (such as primality) on the choice of the small multiplier  $j$  which make it somewhat less likely that faulty computations will remain undetected. Another modification of the technique is to test each half of the computation by a separate recomputation modulo a different small modulus, rather than by comparing the two results modulo a common small modulus. More precisely, the implementation can choose two small numbers  $j_1$  and  $j_2$ , and then compute the following quantities:

$$v_1 = x \pmod{j_1 * p}, v_2 = x \pmod{j_2 * q}, v_3 = x \pmod{j_1}, v_4 = x \pmod{j_2};$$

$$d_1 = d \pmod{\phi(j_1 * p)}, d_2 = d \pmod{\phi(j_2 * q)}, d_3 = d \pmod{\phi(j_1)}, d_4 = d \pmod{\phi(j_2)};$$

$$w_1 = v_1^{d_1} \pmod{j_1 * p}, w_2 = v_2^{d_2} \pmod{j_2 * p}, w_3 = v_3^{d_3} \pmod{j_1}, w_4 = v_4^{d_4} \pmod{j_2}.$$

To check the correctness of the computation, the black box verifies that  $w_1 = w_3 \pmod{j_1}$  and  $w_2 = w_4 \pmod{j_2}$ . The only expensive operations are the computation of  $w_1$  and  $w_2$ , since the small exponentiations in the computation of  $w_3$  and  $w_4$  are very efficient. This recomputation technique is slower than the original comparison technique, but may be slightly more resistant to certain types of non-random faults.

While the invention has been described with respect to certain embodiments thereof, it will be appreciated by one skilled in the art that variations and modifications may be made without departing from the spirit and scope of the invention.



### BIBLIOGRAPHY

1. E. Biham and A. Shamir, "A New Cryptanalytic Attack on DES", technical report, 10/18/96. An Extended version appears in the Proceedings of Crypto 97; August 1997.
2. D. Boneh, R.A. Demillo and R.J. Lipton, "On the Importance of Checking Computations", technical report, 9/25/96. An Extended version appears in the Proceedings of Eurocrypt 97, May 1997.
3. P. Kocher, "Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks", technical report, 12/7/95.

WHAT IS BEING CLAIMED IS

1. In a method of implementing public key schemes containing the non-CRT form of the modular exponentiation operation  $x^d \pmod{n}$ , for the purpose of making them more resistant to timing attacks, the improvement comprising the steps of:

- a. computing or storing the computed value of  $t = \phi(n)$ , where  $\phi$  is Euler's totient function of the modulus  $n$ ;
- b. selecting some secret integer  $i$ ; and
- c. replacing the computation of  $x^d \pmod{n}$  by the computation of  $x^{(d+i*t)} \pmod{n}$ .

2. In the method of claim 1, the improvement where  $i$  is chosen as a random number in the range  $[0, (2^k)-1]$  for some  $k$ .

3. In the method of claim 2, the improvement where  $k=32$ .

4. In a method of implementing public key schemes containing the CRT form of the modular exponentiation operation  $x^d \pmod{n}$  where  $n=p*q$ , for the purpose of making them more resistant to timing and fault attacks, the improvement comprising the steps of:

- a. selecting some secret integer  $j$ ;
- b. computing  $v_1 = x \pmod{j*p}$ ,  $v_2 = x \pmod{j*q}$ ,  $d_1 = d \pmod{\phi(j*p)}$ ,  $d_2 = d \pmod{\phi(j*q)}$ ,  $w_1 = v_1^{d_1} \pmod{j*p}$ , and  $w_2 = v_2^{d_2} \pmod{j*q}$ ;
- c. aborting the computation if  $w_1$  and  $w_2$  are not equal modulo  $j$ ;

- d. otherwise, computing  $y_1 = w_1 \pmod{p}$ ,  $y_2 = w_2 \pmod{q}$ , and combining them by the Chinese Remainder Theorem to obtain the result of  $x^d \pmod{n}$ .
5. In the method of claim 4 improvement where  $j$  is chosen as a prime number.
6. In the method of claim 4 improvement where  $j$  is chosen as a random number in the range  $[0, (2^k) - 1]$  for some  $k$ .
7. In the method of claim 5 improvement where  $k = 32$ .
8. In a method of implementing public key schemes containing the CRT form of the modular exponentiation operation  $x^d \pmod{n}$  where  $n = p * q$ , for the purpose of making them more resistant to timing and fault attacks, the improvement comprising the steps of:
  - a. selecting two secret integers  $j_1$  and  $j_2$ ;
  - b. computing  $v_1 = x \pmod{j_1 * p}$ ,  $v_2 = x \pmod{j_2 * q}$ ,  $d_1 = d \pmod{\phi(j_1 * p)}$ ,  $d_2 = d \pmod{\phi(j_2 * q)}$ ,  $w_1 = v_1^{d_1} \pmod{j_1 * p}$ , and  $w_2 = v_2^{d_2} \pmod{j_2 * q}$ ;
  - c. computing  $v_3 = x \pmod{j_1}$ ,  $v_4 = x \pmod{j_2}$ ,  $d_3 = d \pmod{j_1}$ ,  $d_4 = d \pmod{j_2}$ ,  $w_3 = v_3^{d_3} \pmod{j_1}$ , and  $w_4 = v_4^{d_4} \pmod{j_2}$ ;
  - d. aborting the computation if  $w_3$  is not equal to  $w_1$  modulo  $j_1$ , or if  $w_4$  is not equal to  $w_2$  modulo  $j_2$ ; d. otherwise, computing  $y_1 = w_1 \pmod{p}$ ,  $y_2 = w_2 \pmod{q}$ , and combining them by the

Chinese Remainder Theorem to obtain the result of  $x^d \pmod n$ .

9. In the method of claim 8 improvement where  $j_1$  and  $j_2$  are prime numbers.
10. In the method of claim 8 improvement where  $j_1$  and  $j_2$  are chosen as random numbers in the range  $[0, (2^k)-1]$  for some  $k$ .
11. In the method of claim 10 improvement where  $k=32$ .
12. In apparatus for implementing public key schemes containing the non-CRT form of the modular exponentiation operation  $x^d \pmod n$ , for the purpose of making them more resistant to timing attacks, the improvement comprising:
  - a. means for computing or storing the computed value of  $t=\phi(n)$ , where  $\phi$  is Euler's totient function of the modulus  $n$ ;
  - b. means for selecting some secret integer  $i$ ; and
  - c. means for replacing the computation of  $x^d \pmod n$  by the computation of  $x^{(d+i*t)} \pmod n$ .
13. In apparatus according to claim 1, the improvement where  $i$  is chosen as a random number in the range  $[0, (2^k)-1]$  for some  $k$ .
14. In apparatus according to claim 2, the improvement where  $k=32$ .
15. In apparatus for implementing public key schemes containing the CRT form of the modular exponentiation operation  $x^d \pmod n$  where  $n=p*q$ , for the purpose of making them more resistant to timing and fault attacks, the improvement comprising:
  - a. means for selecting some secret integer  $j$ ;

- b. means for computing  $v_1 = x \pmod{j \cdot p}$ ,  $v_2 = x \pmod{j \cdot q}$ ,  $d_1 = d \pmod{\phi(j \cdot p)}$ ,  $d_2 = d \pmod{\phi(j \cdot q)}$ ,  $w_1 = v_1^{d_1} \pmod{j \cdot p}$ , and  $w_2 = v_2^{d_2} \pmod{j \cdot q}$ ;
  - c. means for aborting the computation if  $w_1$  and  $w_2$  are not equal modulo  $j$ ;
  - d. otherwise, means for computing  $y_1 = w_1 \pmod{p}$ ,  $y_2 = w_2 \pmod{q}$ , and combining them by the Chinese Remainder Theorem to obtain the result of  $x^d \pmod{n}$ .
16. In apparatus according to claim 4, the improvement where  $j$  is chosen as a prime number.
17. In apparatus according to claim 4, the improvement where  $j$  is chosen as a random number in the range  $[0, (2^k) - 1]$  for some  $k$ .
18. In apparatus according to claim 5 the improvement where  $k = 32$ .
19. In apparatus for implementing public key schemes containing the CRT form of the modular exponentiation operation  $x^d \pmod{n}$  where  $n = p \cdot q$ , for the purpose of making them more resistant to timing and fault attacks, the improvement comprising:
- a. selecting two secret integers  $j_1$  and  $j_2$ ;
  - b. computing  $v_1 = x \pmod{j_1 \cdot p}$ ,  $v_2 = x \pmod{j_2 \cdot q}$ ,  $d_1 = d \pmod{\phi(j_1 \cdot p)}$ ,  $d_2 = d \pmod{\phi(j_2 \cdot q)}$ ,  $w_1 = v_1^{d_1} \pmod{j_1 \cdot p}$ , and  $w_2 = v_2^{d_2} \pmod{j_2 \cdot q}$ ;
  - c. computing  $v_3 = x \pmod{j_1}$ ,  $v_4 = x \pmod{j_2}$ ,  $d_3 = d \pmod{j_1}$ ,

$d_4 = d \pmod{j_2}$ ,  $w_3 = v_3^{d_3} \pmod{j_1}$ , and  $w_4 = v_4^{d_4} \pmod{j_2}$ ;

- d. aborting the computation if  $w_3$  is not equal to  $w_1$  modulo  $j_1$ , or if  $w_4$  is not equal to  $w_2$  modulo  $j_2$ ; d. otherwise, computing  $y_1 = w_1 \pmod{p}$ ,  $y_2 = w_2 \pmod{q}$ , and combining them by the Chinese Remainder Theorem to obtain the result of  $x^d \pmod{n}$ .

20. In apparatus according to claim 8, the improvement where  $j_1$  and  $j_2$  are prime numbers.

21. In apparatus according to claim 8, the improvement where  $j_1$  and  $j_2$  are chosen as random numbers in the range  $[0, (2^k) - 1]$  for some  $k$ .

22. In apparatus according to claim 10, the improvement where  $k=32$ .

1/2

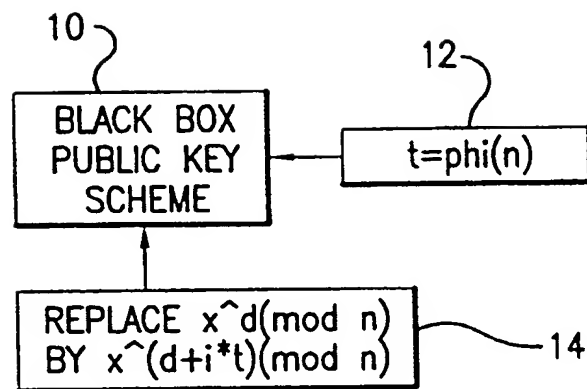


FIG. 1

2/2

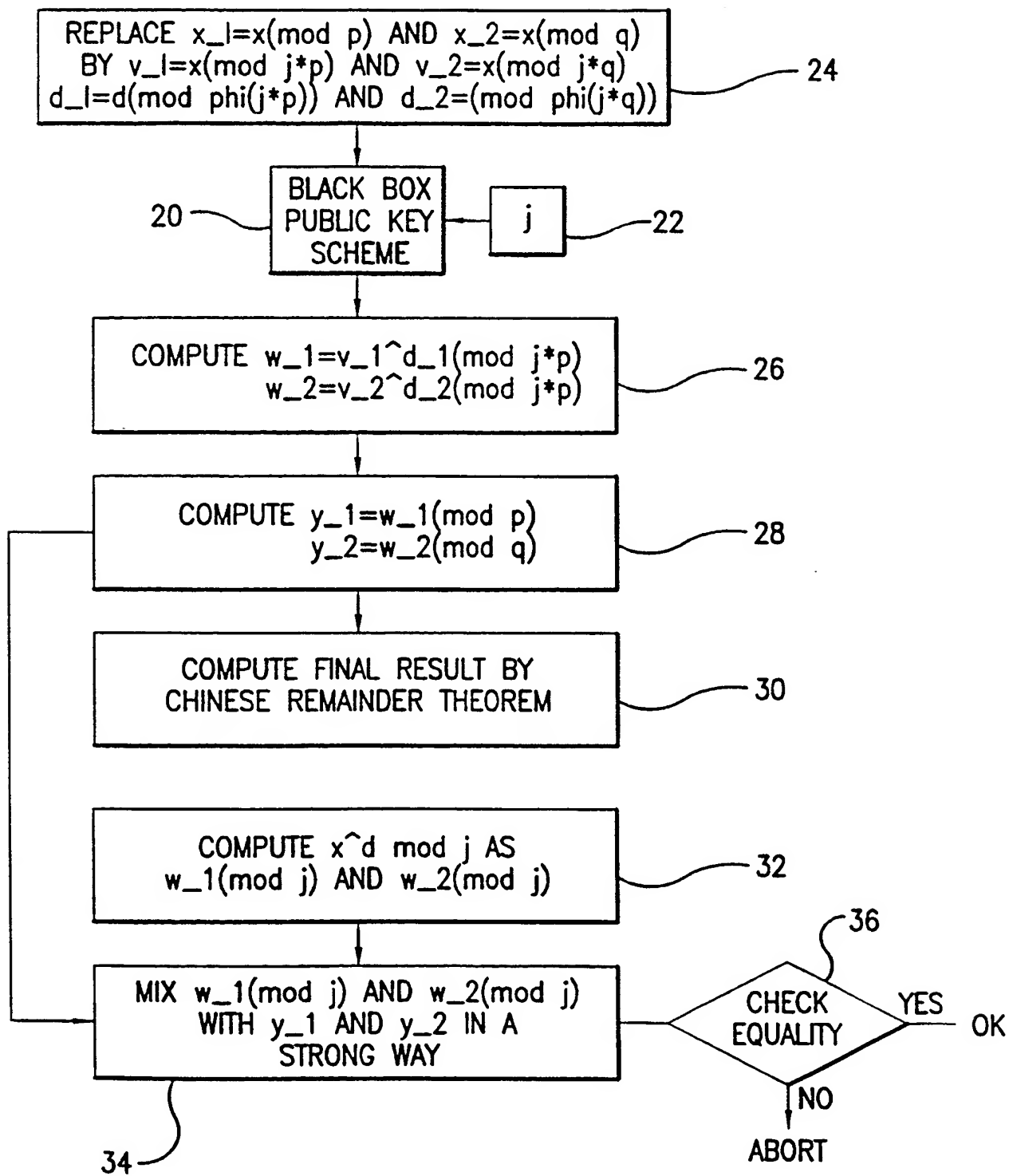


FIG. 2



# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 98/09593

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 6 H04L9/30

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KOCHER P C: "TIMING ATTACKS ON IMPLEMENTATIONS OF DIFFIE-HELLMAN, RSA, DSS, AND OTHER SYSTEMS" ADVANCES IN CRYPTOLOGY - CRYPTO '96, 16TH. ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE SANTA BARBARA, AUG. 18 - 22, 1996. PROCEEDINGS, no. CONF. 16, 18 August 1996, pages 104-113, XP000626590	1, 12
A	KOBLITZ N (ED ) BERLIN (DE) see page 109, line 17 - page 110, line 15  see page 111, line 23 - page 112, line 17 see page 112, line 35 - line 37 --- -/--	4, 8, 15, 19



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

**Special categories of cited documents:**

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

29 September 1998

Date of mailing of the international search report

06/10/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Holper, G

# INTERNATIONAL SEARCH REPORT

Int. l. Application No

PCT/US 98/09593

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>EP 0 682 327 A (YEDA RESEARCH)  15 November 1995  see abstract  see page 4, line 6 - line 26  -----</p>	<p>1,4,8,  12,15,19</p>

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 98/09593

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 682327 A	15-11-1995	US 5504817 A	02-04-1996

This Page Blank (uspto)